

BOTS OF WAR

Dokumentacja techniczna

Komunikacja gra - bot

W ramach rozgrywki grające boty oczekują na dane z serwera, na podstawie których interpretują aktualny stan gry. Serwer wymaga odpowiedzi ze strony bota, który wysyła do serwera gry obiekt rozkazów.

Elementy obiektu:

- **mapPath** - ścieżka do mapy na której rozgrywana jest gra.
- **gameId** - id rozgrywanej gry.
- **tournamentId** - opcjonalnie id turnieju.
- **currentTurn** - aktualna tura.
- **currentStep** - aktualny 'krok' gry (każda tura dzieli się na dwa kroki – gracza 1 i gracza 2).
- **player1** - obiekt gracza 1.
- **player2** - obiekt gracza 2.
- **mines** - lista kopalń z ich danymi.
- **lastActions** - obiekt ostatniej wykonanej akcji w grze.
- **lastLogs** - ostatnie zanotowane przez backend logi.

Metody:

- **getMapPath()** - pobiera ścieżkę do mapy.
- **getGameId()** - pobiera id gry.
- **getTournamentId()** - pobiera id turnieju.
- **getCurrentTurn()** - pobiera aktualny turę gry.
- **getCurrentStep()** - pobiera aktualny krok gry.
- **getPlayer1()** - pobiera obiekt pierwszego gracza.
- **getPlayer2()** - pobiera obiekt drugiego gracza.
- **getMines()** - pobiera listę kopalń.

Określenie aktywnego gracza i interpretacja stanu gry:

Aby bot mógł określić dla którego gracza ma wykonać akcję powinien odnieść się do stanu obiektów typu Player otrzymanych w obiekcie GameState z serwera gry w następujący sposób:

- **isActive()** - zwraca wartość **true** dla aktywnego gracza lub **false** dla nieaktywnego.
- **getUnits()** - zwraca listę jednostek gracza.
- **getGold()** - zwraca aktualną ilość złota.
- **getId()** - zwraca id gracza.

(UWAGA!) Gra jest otwarta i każdy ruch przeciwnika jest widoczny. Dlatego też mamy możliwość pobierania danych na temat każdej jednostki przeciwnika.

Dane własnej bazy lub bazy przeciwnika pobieramy:

- **getBase()** - zwraca obiekt typu Base.
- **getId()** - zwraca id jednostki.
- **getCoordinates()** - zwraca (Coordinates) współrzędne bazy.
 - **getX()** - zwraca (int) współrzędną x.
 - **getY()** - zwraca (int) współrzędną y.
- **getOwner()** - zwraca właściciela bazy.

Dane jednostek (Unit):

- **getName()** - zwraca (String) nazwę jednostki.
- **getId()** - zwraca id jednostki.
- **getCoordinates()** - zwraca (Coordinates) współrzędne jednostki.
 - **getX()** - zwraca (int) współrzędną x.
 - **getY()** - zwraca (int) współrzędną y.
- **getOwner()** - zwraca właściciela jednostki.
- **getCost()** - zwraca (int) koszt wykonania jednostki.
- **getHp()** - zwraca (int) ilość życia jednostki.
- **getRangeOfAttack()** - zwraca (int) zasięg ataku jednostki.
- **getActionPoints()** - zwraca (int) ilość możliwych ruchów do wykonania.
- **isEntrench()** - zwraca (boolean) czy dana jednostka jest okopana.
- **getDamage()** - zwraca (int) siłę ataku jednostki.

Przygotowanie rozkazu:

Poprawny obiekt rozkazu (Order) składa się z listy akcji. Poszczególne akcje zawiera 3 parametry określone przez gracza (bota). Obiekt rozkazu może zawierać pustą listę akcji, co jest równoznaczne z pominięciem tury.

Konstruktor obiektu Action zawiera poszczególne parametry:

- id (string) - id własnej jednostki lub bazy.
- actionType - typ akcji (ATTACK, RECRUIT, MOVE, ENTRENCH).
- target (string) - w zależności od typu akcji:
 - ATTACK: id atakowanej jednostki przeciwnika.
 - RECRUIT: symbol określający typ rekrutowanej jednostki:
 - "W" - worker;
 - "H" - horseman;
 - "F" - warrior;
 - "A" - archer.
 - MOVE: kierunek ruchu określony symbolem:
 - "U" - góra;
 - "D" - dół;
 - "L" - lewo;
 - "R" - prawo.
 - ENTRENCH: id tej samej wybranej jednostki.